Current IMS Database

Segment
Prefix & Data   20

DS Group

Fig 1A
(Prior Art)

Invention Database

Segment   22
Prefix
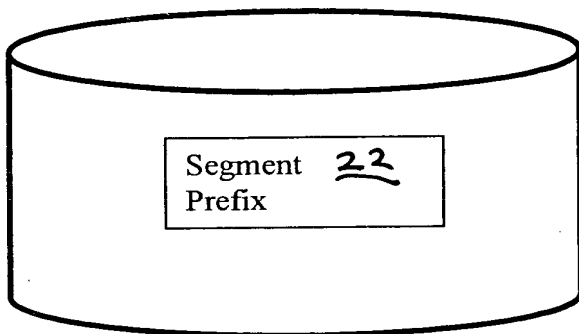
Segment   24
Data
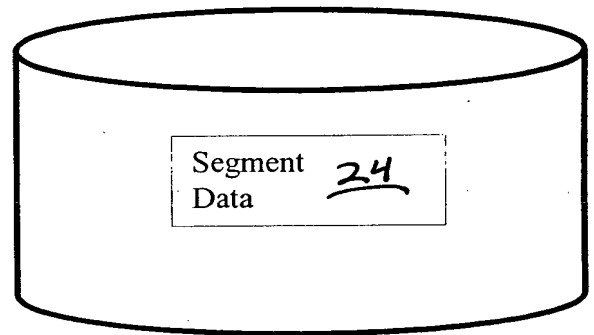
Directory DS

Segdata DS

Fig 1 B

Layout of Segment in Directory Dataset

Segment Prefix	26	Segment Data		28
			Metadata	
Seg Code & Delete Byte **20**	Prefix Pointers **32**	Pointer to Seg Data **34**	Seg Key **38**	Born-On-Date **36**

Figure 2A. Split Segment Composition – Prefix Portion with Metadata in segment data portion

Layout of Segment in Segdata Dataset

Segment Prefix		26		Seg Data 28
30 Seg Code & Delete Byte	Prefix Pointers **32**	Metadata		**34** Pointer to Seg Data
		Seg Key **38**	Born-On-Date **36**	

Figure 2B. Split Segment Composition – Prefix Portion with Metadata in segment prefix portion

Layout of Segment in Segdata Dataset

Segment Prefix 40	Segment Data 42	Trans-parent 44
Seg code & delete byte 46	User Data 48	Born on Date 50

Fig. 3

```
DBD      NAME=IVPDB1,ACCESS=(HIDAM,OSAM)                              122

DIR      DD1=DFSIVD1,SIZE=2048,UOW=(500,50,10)        ←

DATASET DD1=DFSIVD1A,DEVICE=3380,SIZE=2048
SEGM    NAME=A1111111,PARENT=0,BYTES=40,RULES=(LLV,LAST),PTR=(TB,CTR)
FIELD   NAME=(A1111111,SEQ,U),BYTES=010,START=00001,TYPE=C
FIELD   NAME=A9999999,BYTES=010,START=00011,TYPE=C
LCHILD  NAME=(A1,IVPDB1I),POINTER=INDX,RULES=LAST
LCHILD  NAME=(A1X,IVPDB1X),POINTER=INDX
XDFLD   NAME=AXXXXXXX,SEGMENT=A1111111,SRCH=(A9999999)
LCHILD  NAME=(C1X,IVPDB1Z),POINTER=INDX
XDFLD   NAME=CXXXXXXX,SEGMENT=C1111111,SRCH=(C9999999)

DATASET DD1=DFSIVD1B,DEVICE=3380,SIZE=4096
SEGM    NAME=B1111111,PARENT=A1111111,BYTES=(1000,50),              X
            RULES=(LLV,LAST),PTR=(TB)
FIELD   NAME=(B1111111,SEQ,M),BYTES=010,START=00003,TYPE=C
FIELD   NAME=/SXB1
LCHILD  NAME=(B1X,IVPDB1Y),POINTER=INDX
XDFLD..NAME=BXXXXXXX,SEGMENT=B1111111,SRCH=(B1111111),SUBSEQ=(/SXB1)

DATASET DD1=DFSIVD1C,DEVICE=3380,SIZE=8192
SEGM    NAME=C1111111,PARENT=B1111111,COMPRTN=(DFSKMPX0,DATA,INIT),  X
            RULES=(LLV,LAST),PTR=(TB),BYTES=(8000,50)
FIELD   NAME=(C1111111,SEQ,U),BYTES=010,START=00003,TYPE=C
FIELD   NAME=C9999999,BYTES=010,START=00011,TYPE=C

DIRGEN

DBDGEN
FINISH
END
```

Figure 4A Sample HIDAM DBD

```
DBD       NAME=IVPDB2,ACCESS=HDAM,RMNAME=(DFSHDC40,4,1000)              124

DIR   DD1=DFSIVD2,UOW=(100,10)

DATASET DD1=DFSIVD2A,DEVICE=3380,SIZE=2048
SEGM     NAME=A1111111,PARENT=0,BYTES=40,RULES=(LLL,LAST),              X
            COMPRTN=(DFSKMPX0,DATA,INIT)
FIELD    NAME=(A1111111,SEQ,U),BYTES=010,START=00001,TYPE=C


DATASET DD1=DFSIVD2B,DEVICE=3380,SIZE=4096
SEGM     NAME=B1111111,PARENT=A1111111,BYTES=(1000,50),                 X
            RULES=(LLV,LAST),PTR=(TB)
FIELD    NAME=(B1111111,SEQ,U),BYTES=010,START=00003,TYPE=C


DATASET DD1=DFSIVD2C,DEVICE=3380,SIZE=8192
SEGM     NAME=C1111111,PARENT=B1111111,COMPRTN=(DFSKMPX0,DATA,INIT),
         RULES=(LLV,LAST),PTR=TB,BYTES=8000
FIELD    NAME=(C1111111,SEQ,U),BYTES=010,START=00001,TYPE=C

DIRGEN

DBDGEN
FINISH
END
```

Figure 4β Sample HDAM DBD

Secondary Index

Ptr	Key _58_	Token

Index Record _50_

_56_

_60_

Root Segment _64_

Index _62_
Target Seg

_54_

Target Database
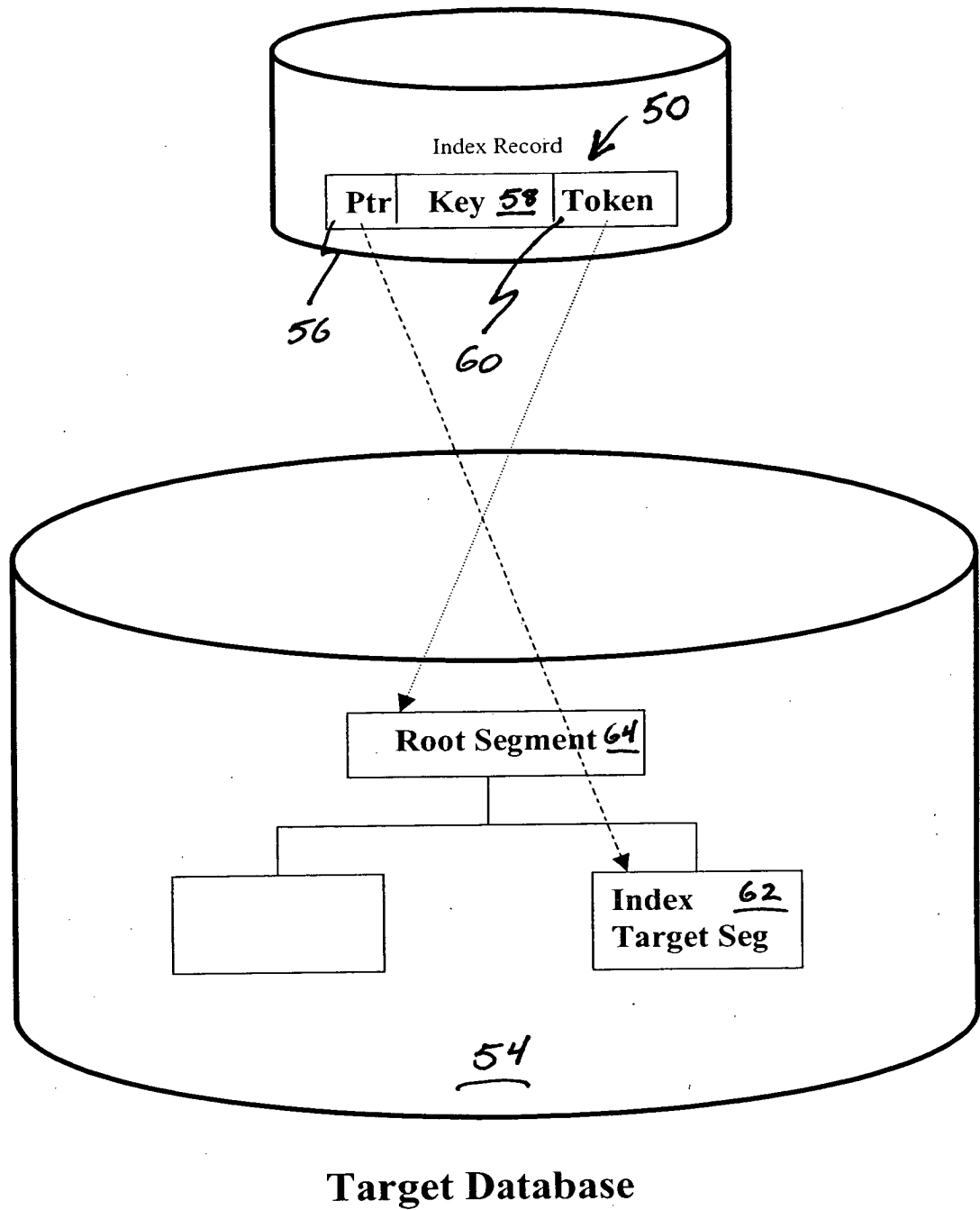
Figure _5_ Secondary Index Architecture

Secondary Index

Index **51** Record		

56   60

58

Ptr	Key	Token

Target **62** Segment			Root **64** Segment	

Block   **66**

54

Target Database
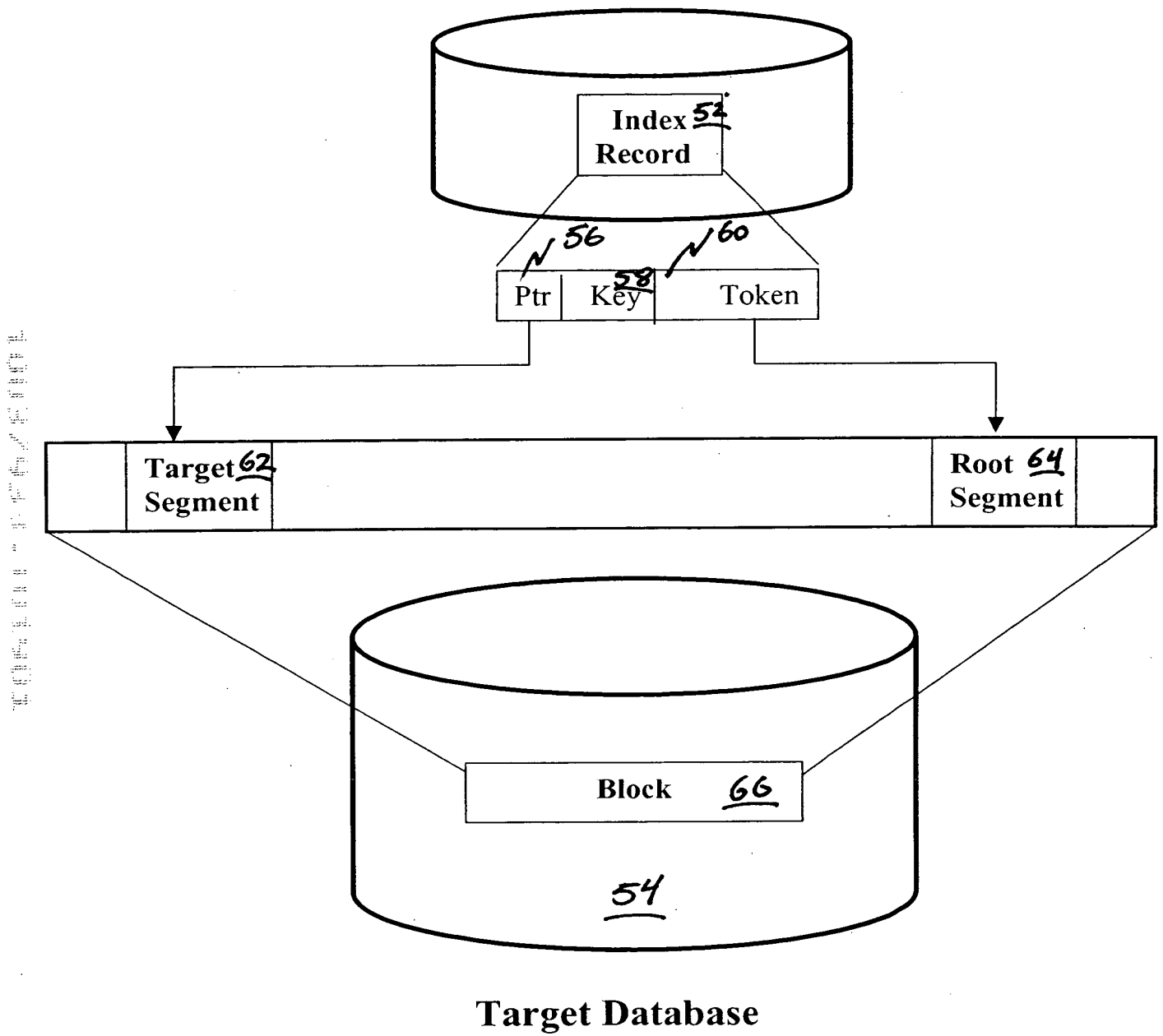
Figure 6 Secondary Index Before Reorganizing

Secondary Index

Index
Record

Ptr | Key | Token

56   *58*   *60*

Target *62*
Segment

Root *64*
Segment

Block   *66*

54
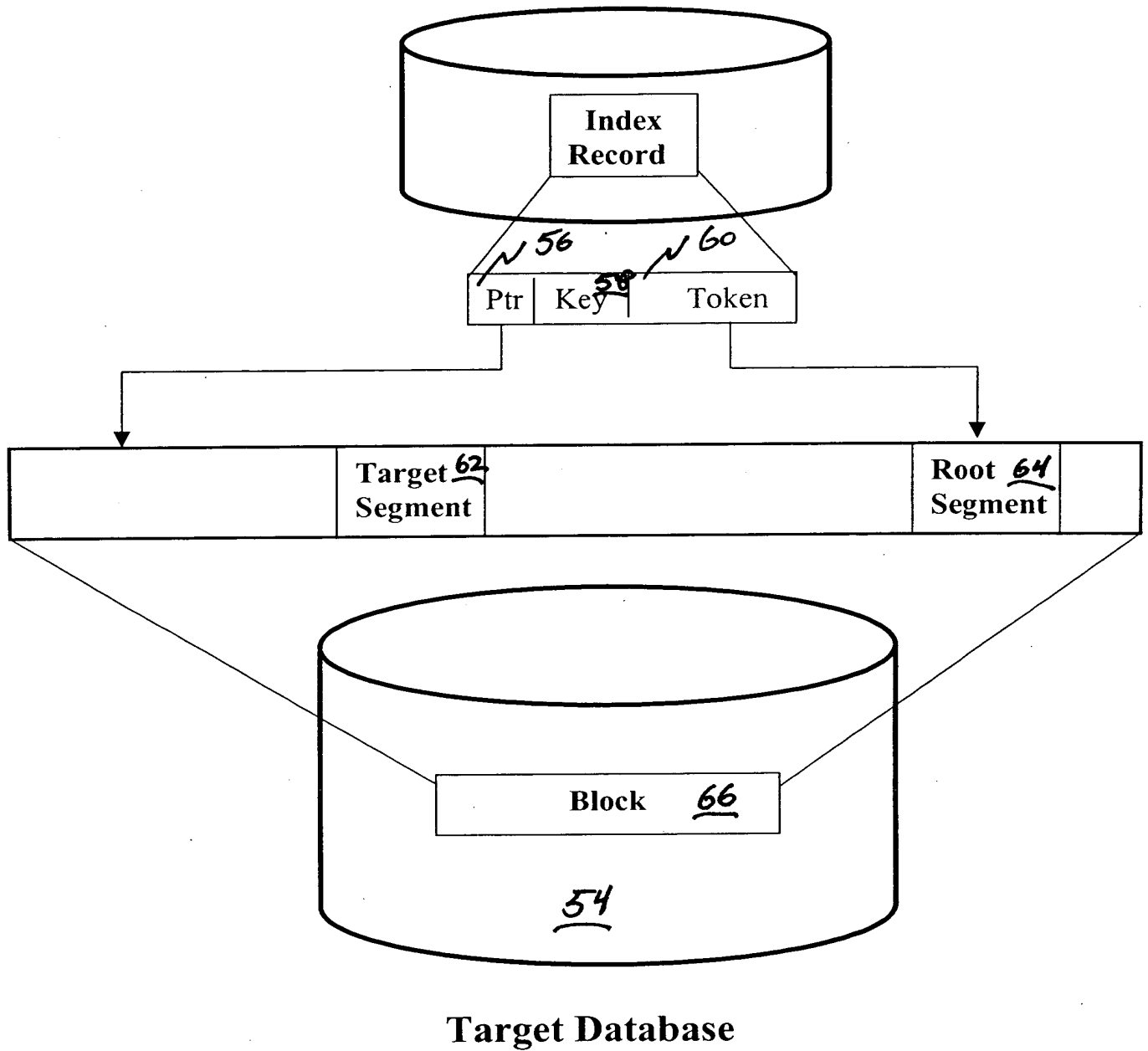
Target Database

Figure 7   Secondary Index After Reorganizing

```
                                    ●  ⌐

                                    ●  ⌐

                            ╭──────────────╮
                            ╰──────────────╯
                                    │
                                    ▼
            ┌───────────────────────────────────────────┐
            │        IMS extracts secondary index key    │
            │           from A/P's Read request           │
            │                    68                        │
            └───────────────────────────────────────────┘
                                    │
                                    ▼
            ┌───────────────────────────────────────────┐
            │        IMS extracts Index Record from       │
            │          secondary index dataset with       │
            │            A/P's requested key              │
            │                    70                        │
            └───────────────────────────────────────────┘
                                    │
                                    ▼
            ┌───────────────────────────────────────────┐
            │  IMS extracts Direct Pointer from Index     │
            │   Record and attempts to read Target        │
            │   Segment at that location in the Target    │
            │        Database      72                      │
            └───────────────────────────────────────────┘
                                    │
                                    ▼
```

Valid
Target Segment
found?
74

NO

Invoke Invention Software
To Correct Direct Pointer
in Secondary Index record.
76

Figure 8 Retrieving a Target Segment via a Secondary Index

```
         ●       .-           ●       .-
                          ⌈          ⌉
                          ⌊_____⌋
                               │
                               ▼
         ┌─────────────────────────────────────────┐
         │   Invention software extracts Token      │
         │     From Secondary Index record          │
         │                 78                        │
         └─────────────────────────────────────────┘
                               │
                               ▼
         ┌─────────────────────────────────────────┐
         │     Extracts Root Segment's RBA          │
         │        And Target Segment's              │
         │        Born-On-Data from Token           │
         │                 80                        │
         └─────────────────────────────────────────┘
                               │
                               ▼
         ┌─────────────────────────────────────────┐
         │   Requests IMS to read Root Segment      │
         │       In Target Database using           │
         │           Provided RBA                   │
         │                 82                        │
         └─────────────────────────────────────────┘
                               │
                               ▼
         ┌─────────────────────────────────────────┐
         │     Follow hierarchical pointers         │
         │       Within the Root Segment            │
         │     Searching for a Target Segment       │
         │     Containing born-on-date    84        │
         └─────────────────────────────────────────┘
                               │
                               ▼
         ┌─────────────────────────────────────────┐
         │       Replace Direct Pointer             │
         │     In Secondary Index record            │
         │      With RBA of Target Segment          │
         │                 86                        │
         └─────────────────────────────────────────┘
                               │
                               ▼
                          ⌈          ⌉
                          ⌊_____⌋
```

Figure 9 Correcting Direct Pointer in a Secondary Index

ACB Library

DMB *90*

Database Load Program *94*

DMB Pool *96*

DMB *98*

Invention
Software *100*

Invention Database

104

DMB *106*

Directory DS *102*
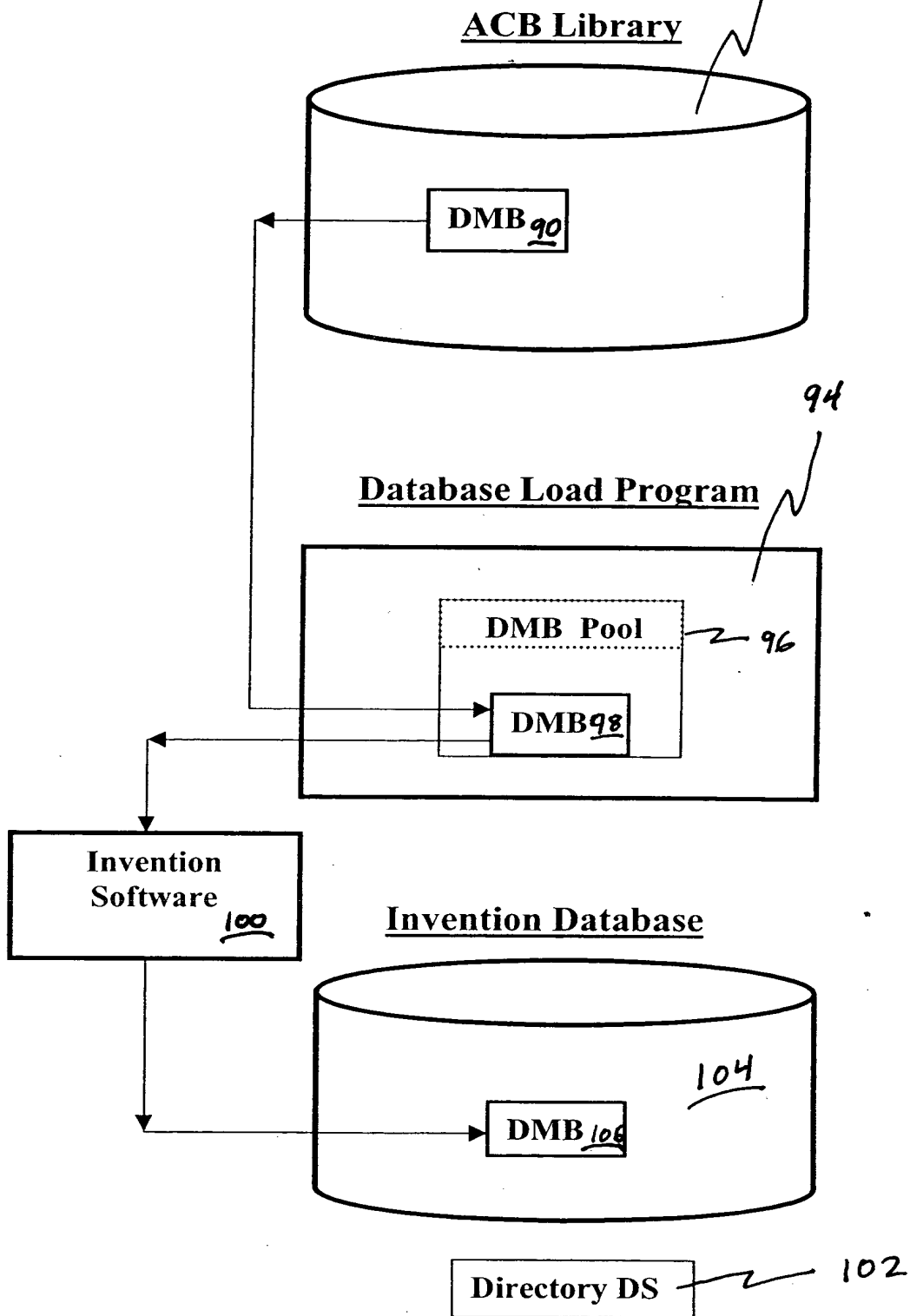
Figure 10 Saving the Database Definition at DB Load Time

ACB Library

DMB 90

Database Processing Program

DMB Pool   96

DMB 98

Invention
Software
100

Invention Database

DMB 104

DMB 106

Directory DS   102
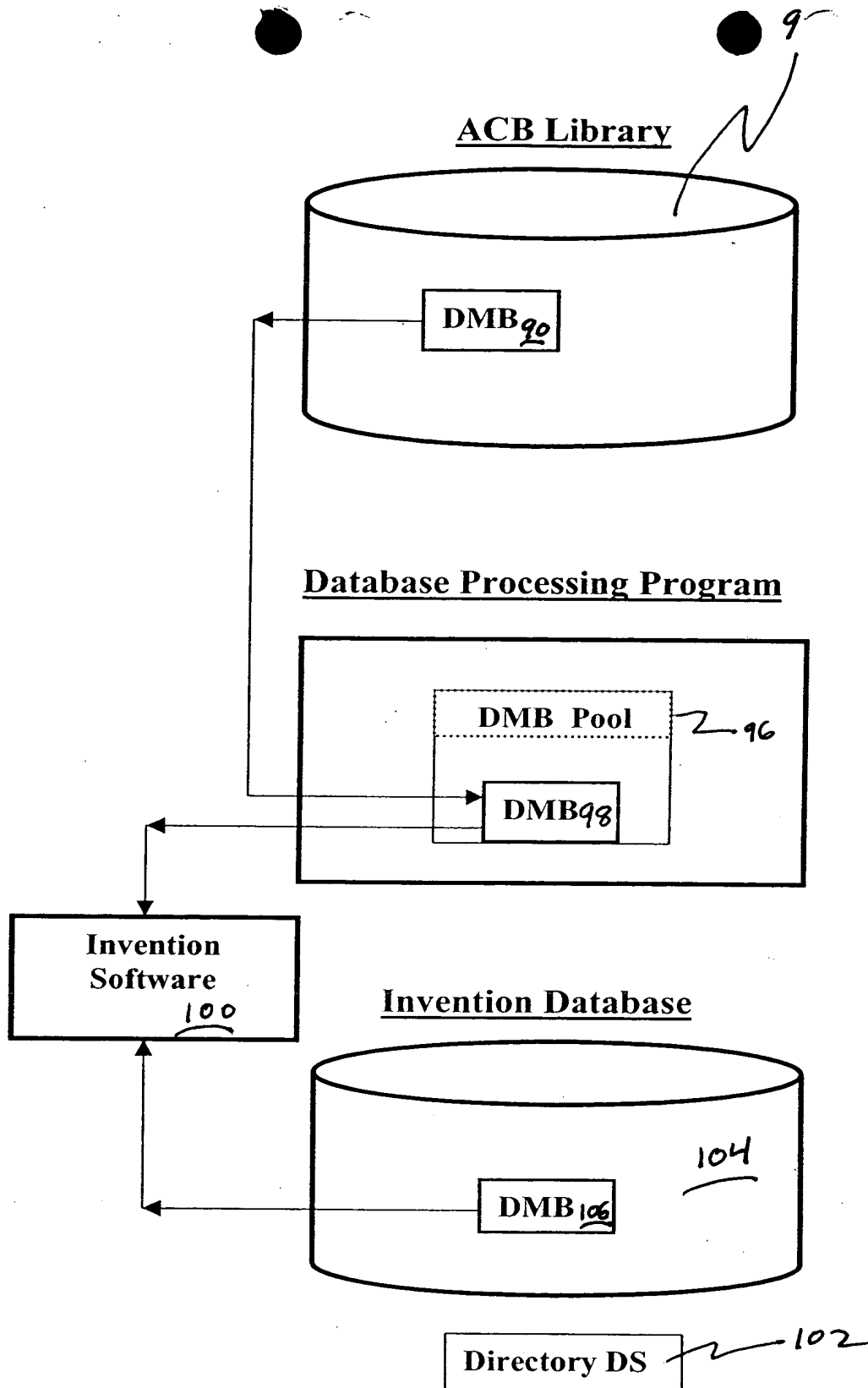
Figure 11 Checking the Database Definition at DB Processing
Time

Data Base Description **126**

Block #1
Block #2
Block #3
Block #4
Block #5
Block #6
Block #7
Block #8
Block #9
Block #10

116

116

110

Prime Area

BLOCK #1
BLOCK #2
BLOCK #3
BLOCK #4
BLOCK #5

118

118

112

Dependent Overflow Area

Block #1

Block #2

120

120

114

Independent Overflow Area
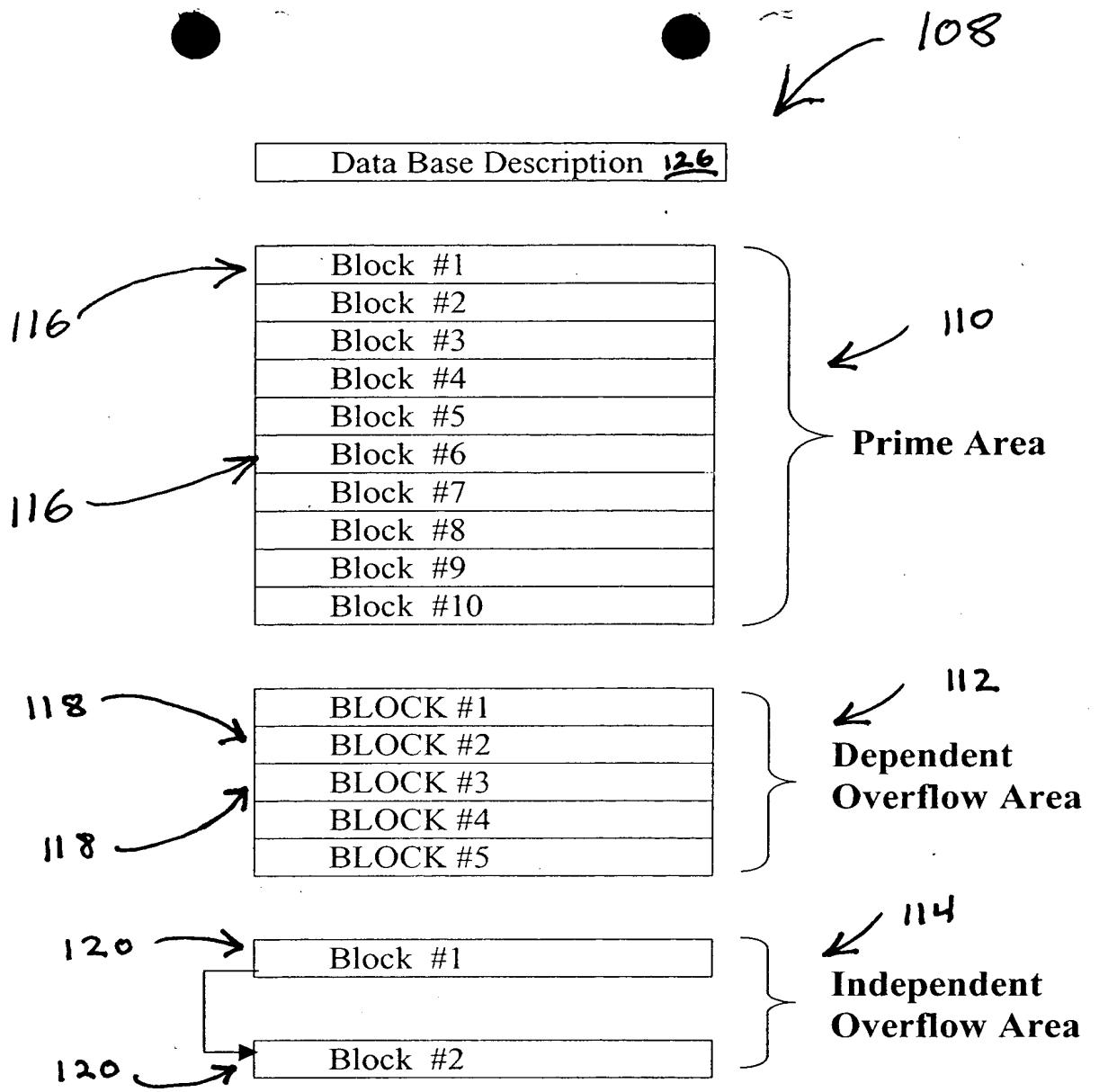
Figure 12. Unit Of Work (UOW) Architecture

132

HDAM RAA

108

108

134

HDAM
Overflow

108

108

| Prime UOW #1 |
| Prime UOW #2 |
| Prime UOW #3 |
| Prime UOW #4 |
| Prime UOW #5 |
| Data Base Description 126 |
| DOVF UOW #1 |
| DOVF UOW #2 |
| DOVF UOW #3 |
| DOVF UOW #4 |
| DOVF UOW. #5 136 |
| Independent Overflow 138 |

128

Prime Area

130

Overflow Area
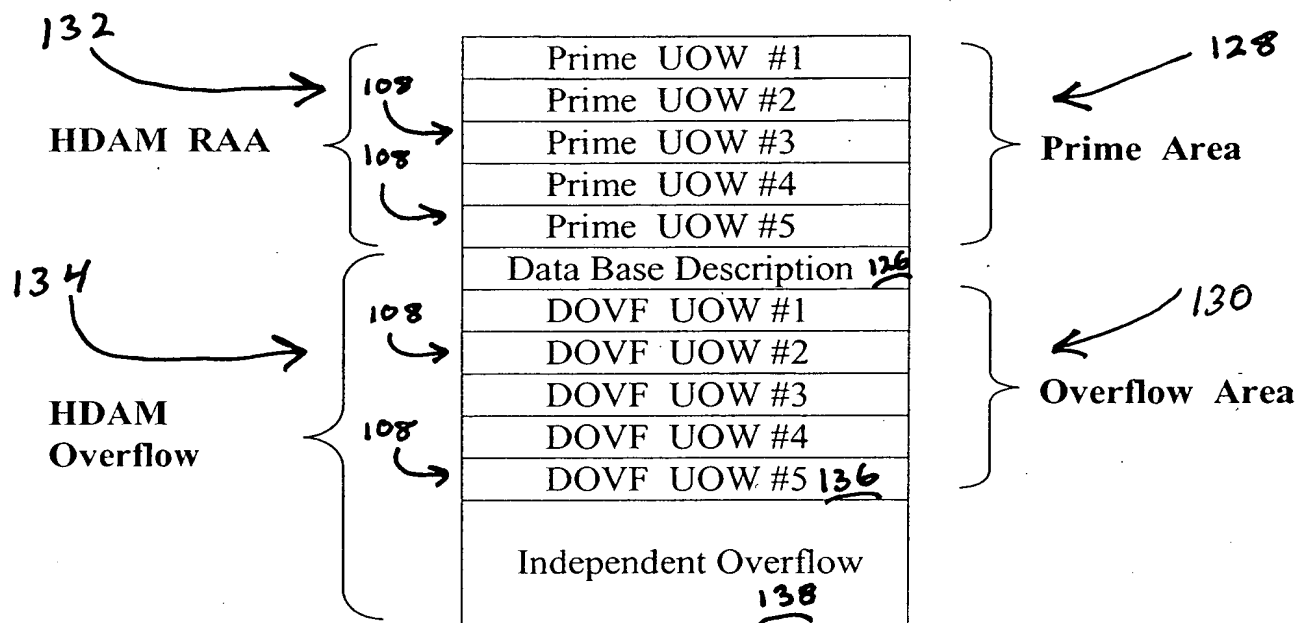
Figure 13. HDAM UOW Architecture

Data Base Description 126
Prime UOW #1
Prime UOW #2
Prime UOW #3
Prime UOW #4
Prime UOW #5
DOVF UOW #1
DOVF UOW #2
DOVF UOW #3
DOVF UOW #4
DOVF UOW #5 136
Independent Overflow 138

108

108

108

108

128

Prime Area

130

Overflow Area
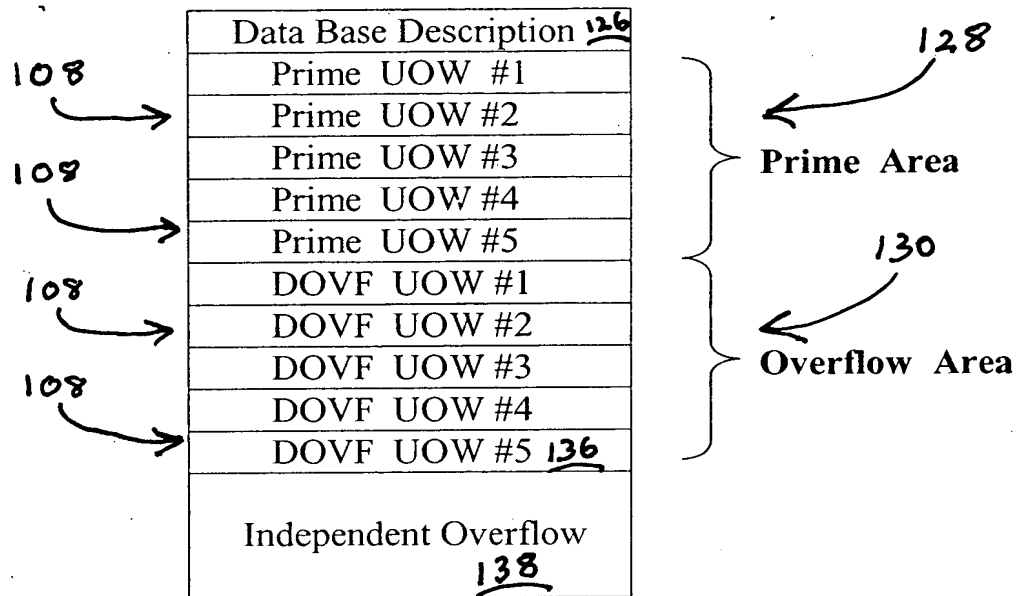
Figure 14. HIDAM UOW Architecture

FSEAP	RAPs	Non-Root Segments	Free Space	Root Segments

140 141 147
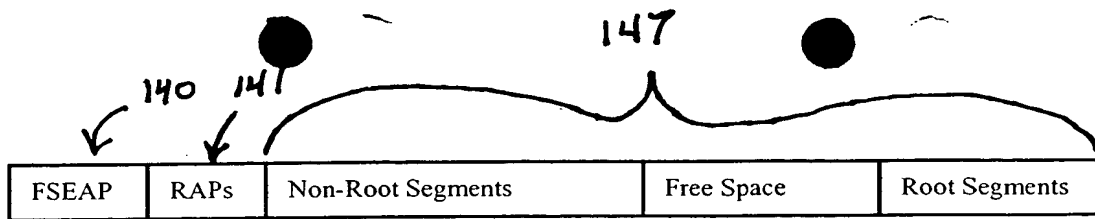
Figure 15. Prime & DOVF Block Composition

FSEAP	UOW Id	Chain	Non-Root Segments	Free Space	Root Segments

140 143 145 147
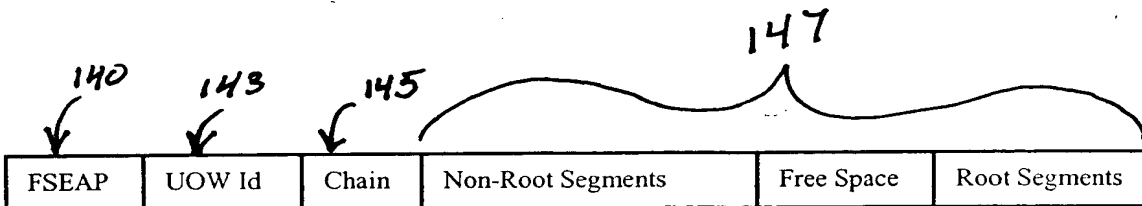
Figure 16. IOVF Block Composition

Root Seg#1	Dep Seg#11	Dep Seg#12	Root Seg#2	Dep Seg#21	Dep Seg#22	Free Space

142 *144* *144* *142* *144* *144* *148*

14
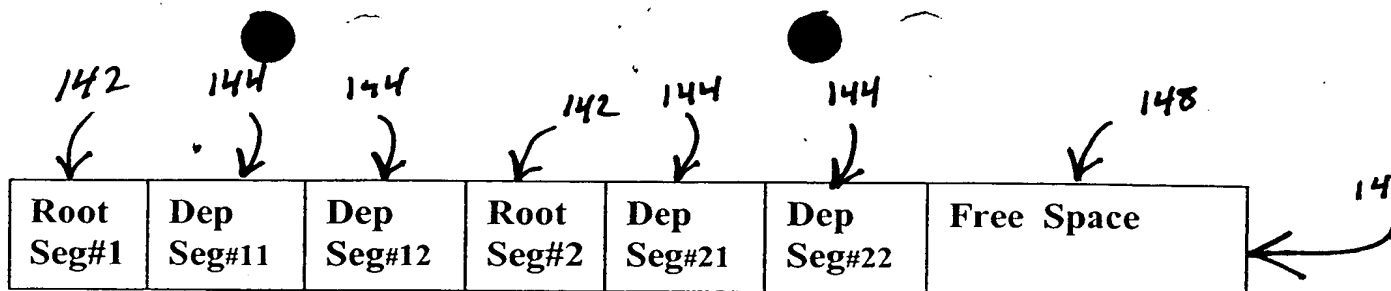
Figure 17 Block Composition Using IMS' Space Management

Dep Seg#11	Dep Seg#12	Dep Seg#21	Dep Seg#22	Free Space	Root Seg#2	Root Seg#1

144 *144* *144* *144* *148* *142* *142*

146
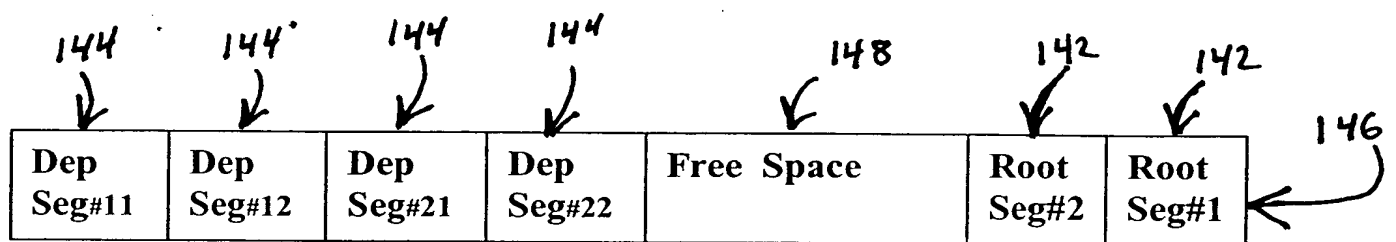
Figure 18 Block Composition Using Invention's Space Management

```
┌─────────────────────────────────┐
│  IMS                        111 │
│─────────────────────────────────│
│  OPEN HD Unload Dataset.        │
│  OPEN database datasets.        │
└─────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────┐          ┌─────────────────────────────────┐
│  Invention                  121 │          │  Invention                  150 │
│─────────────────────────────────│    ┌────▶│─────────────────────────────────│
│ . Obtain address of DMB Control │    │     │  Determine where to place the   │
│  Block and initialize itself.   │    │     │  segment within the block.      │
└─────────────────────────────────┘    │     └─────────────────────────────────┘
              │                         │                   │
              ▼                         │                   ▼
┌─────────────────────────────────┐    │     ┌─────────────────────────────────┐
│  IMS                        131 │    │     │  Invention                  160 │
│─────────────────────────────────│    │     │─────────────────────────────────│
│  Get next user record and build a│   │     │  Insert segment into dictated   │
│  segment (prefix and data).     │    │     │  location.                      │
└─────────────────────────────────┘    │     └─────────────────────────────────┘
              │                         │                   │
              ▼                         │                   ▼
┌─────────────────────────────────┐    │     ┌─────────────────────────────────┐
│  Invention                  141 │    │     │  Invention                  170 │
│─────────────────────────────────│    │     │─────────────────────────────────│
│  Search for a block with space  │────┘     │  Modify IMS Control Blocks      │
│  to contain the segment.        │          │  and return control to IMS.     │
└─────────────────────────────────┘          └─────────────────────────────────┘
```

Figure 19   Space Management at Database Load Time

```
┌─────────────────────────────┐
│  IMS                   210  │
│  ─────────────              │
│  Detects that user is adding a│
│  new segment to database    │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│  IMS                   220  │
│  ─────────────              │
│  Builds a complete segment  │
│  containing prefix and data │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  Invention             230  │          │  Invention             250  │
│  ─────────────              │          │  ─────────────              │
│  Search for a block with space│   ───▶ │  Insert segment into dictated│
│  to contain the new segment │          │  location.                  │
└─────────────────────────────┘          └─────────────────────────────┘
              │                                          │
              ▼                                          ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  Invention             240  │          │  Invention             260  │
│  ─────────────              │          │  ─────────────              │
│  Find space in selected block│         │  Modify IMS Control Blocks  │
│  using invention algorithm  │ ────────▶│  and return control to IMS. │
└─────────────────────────────┘          └─────────────────────────────┘
```
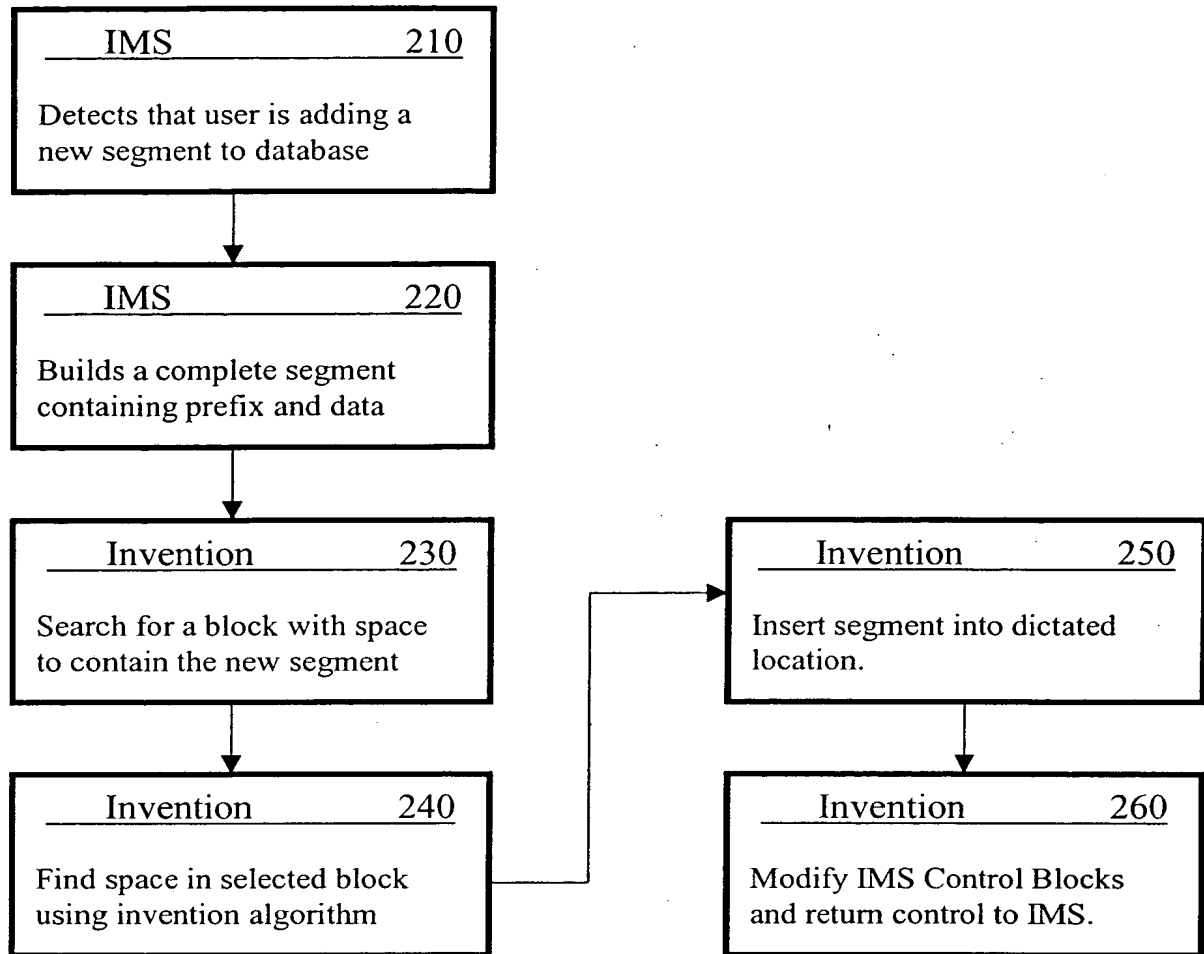
Figure 20 Space Management at Database Update Time

```
┌─────────────────────────────┐        ┌─────────────────────────────┐
│  IMS                   310  │        │  Invention             345  │
│                             │        │  Build a token for the      │
│  OPEN HD Unload Dataset.    │    ┌──▶│  segment.                   │
│  OPEN database datasets.    │    │   │  Split segment and insert   │
│                             │    │   │  token into prefix and data │
└─────────────────────────────┘    │   │  portions.                  │
              │                     │   └─────────────────────────────┘
              ▼                     │                 │
┌─────────────────────────────┐    │                 ▼
│  Invention             320  │    │   ┌─────────────────────────────┐
│                             │    │   │  Invention             350  │
│  Obtain address of DMB      │    │   │  Insert segment with prefix │
│  Control Block and          │    │   │  portion into Directory     │
│  initialize itself.         │    │   │  Dataset, or UOW within     │
│                             │    │   │  this Dataset.              │
└─────────────────────────────┘    │   └─────────────────────────────┘
              │                     │                 │
              ▼                     │                 ▼
┌─────────────────────────────┐    │   ┌─────────────────────────────┐
│  IMS                   330  │    │   │  Invention             360  │
│                             │    │   │  Determine which Segdata    │
│  Get next user record and   │    │   │  Dataset should contain     │
│  build a segment (prefix    │    │   │  user data.                 │
│  and data).                 │    │   │                             │
└─────────────────────────────┘    │   └─────────────────────────────┘
              │                     │                 │
              ▼                     │                 ▼
┌─────────────────────────────┐    │   ┌─────────────────────────────┐
│  Invention             340  │    │   │  Invention             370  │
│  Determine which UOW        │    │   │                             │
│  should contain the         │    │   │  Insert segment data into   │
│  segment with the prefix    │    │   │  dictated Segdata Dataset.  │
│  portion.                   │    │   │                             │
└─────────────────────────────┘    │   └─────────────────────────────┘
              │                     │                 │
              ▼                     │                 ▼
┌─────────────────────────────┐    │   ┌─────────────────────────────┐
│  Invention             342  │    │   │  Invention             380  │
│  Find space in Directory    │    │   │                             │
│  Dataset or selected UOW    │────┘   │  Modify IMS Control Blocks  │
│  in the Dataset;            │        │  and return control to IMS. │
│  Roots go on right side     │        │                             │
│  of block.                  │        └─────────────────────────────┘
└─────────────────────────────┘
```

Figure 21.   Space Management at Database Load Time

```
┌─────────────────────────────┐
│  IMS                    410 │
│                             │
│  Detects that user is adding a
│  new segment to database.   │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  IMS                    420 │          │  Invention              450 │
│                             │          │  Insert segment with prefix │
│  Builds a complete segment  │  ──────▶ │  portion into Directory Dataset,
│  containing prefix and data.│          │  or UOW within this Dataset.│
└─────────────────────────────┘          └─────────────────────────────┘
              │                                        │
              ▼                                        ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  Invention              430 │          │  Invention              460 │
│  Determine which UOW should │          │  Determine which Segdata    │
│  contain the segment with the          │  Dataset should contain the user
│  prefix portion.            │          │  data.                      │
└─────────────────────────────┘          └─────────────────────────────┘
              │                                        │
              ▼                                        ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  Invention              440 │          │  Invention              470 │
│  Find space in Directory Dataset        │                             │
│  or selected UOW in the Dataset;        │  Insert segment data into dictated
│  Roots go on right side of block.       │  Segdata Dataset            │
└─────────────────────────────┘          └─────────────────────────────┘
              │                                        │
              ▼                                        ▼
┌─────────────────────────────┐          ┌─────────────────────────────┐
│  Invention              445 │          │  Invention              480 │
│  Build a token for the segment.         │                             │
│  Split segment and insert token ──────▶ │  Modify IMS Control Blocks  │
│  into prefix and data portions.         │  and return control to IMS. │
└─────────────────────────────┘          └─────────────────────────────┘
```
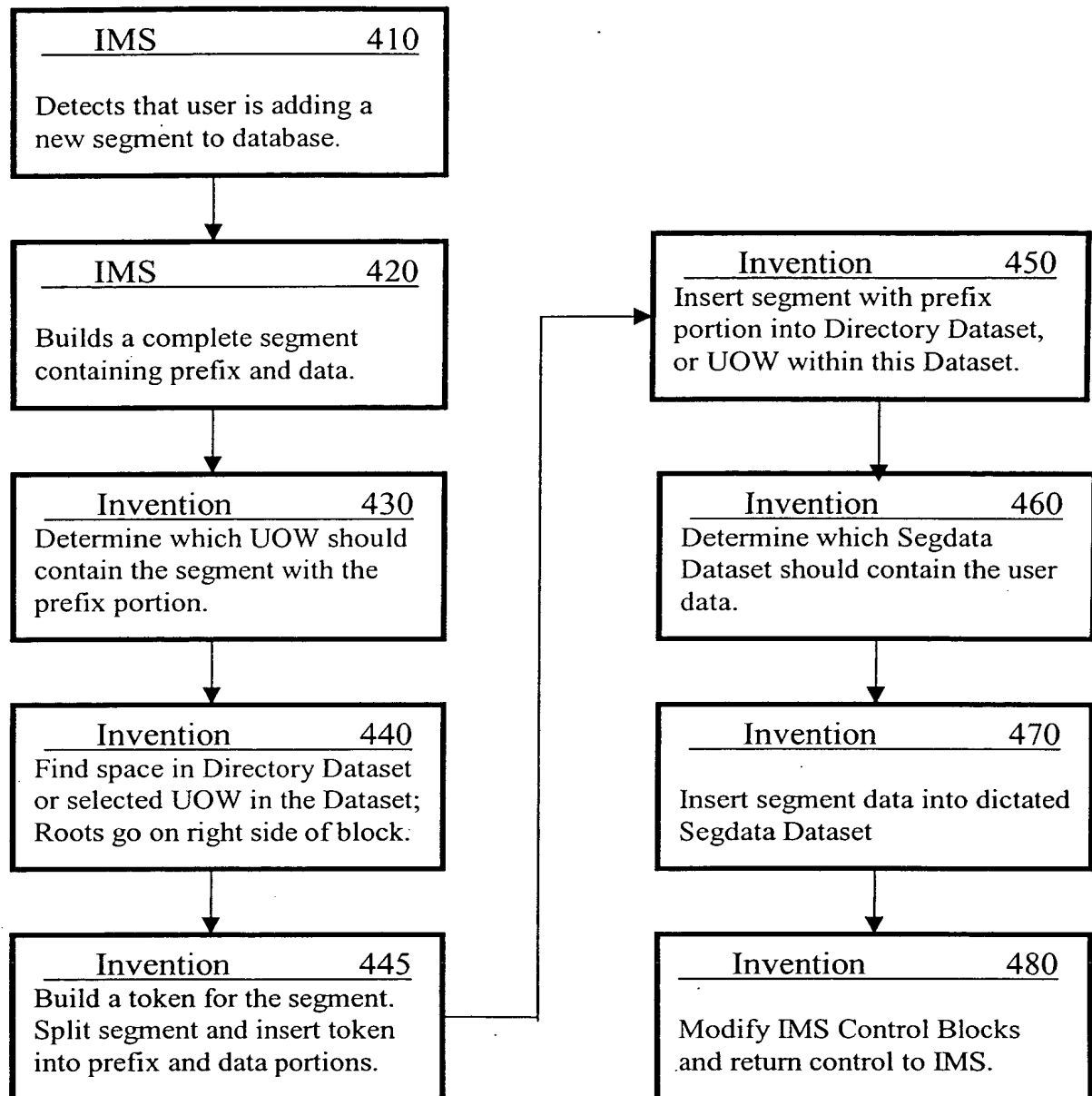
Figure 22. Space Management at Database Update Time